

命令レベル並列演算機構に関する研究

著者	山田 秀和
号	1035
発行年	1985
URL	http://hdl.handle.net/10097/9771

氏 名	やま だ ひで かず 山 田 秀 和
授 与 学 位	工 学 博 士
学位授与年月日	昭和 61 年 3 月 25 日
学位授与の根拠法規	学位規則第 5 条第 1 項
研究科, 専攻の名称	東北大学大学院工学研究科 (博士課程) 情報工学専攻
学 位 論 文 題 目	命令レベル並列演算機構に関する研究
指 導 教 官	東北大学教授 重井 芳治
論 文 審 査 委 員	東北大学教授 重井 芳治 東北大学教授 城戸 健一 東北大学教授 野口 正一 東北大学教授 奈良 久 東北大学助教授 中村 維男

論 文 内 容 要 旨

第 1 章 緒 論

von. Neumann によって提言されたプログラム内蔵方式のノイマン型計算機がこの世に生をうけて40年近く経とうとしている。この間、計算機の応用分野は広がる一方で、今日、計算機は我々の社会生活に不可欠のものとなっている。応用分野の広がりとともに計算機の性能改善には、目覚ましいものがある。しかしながら、原子物理学、気象予測、流体計算などの科学技術計算の分野においてはもちろん、第5世代コンピュータプロジェクトに代表される知識情報処理の分野においても処理速度の高速化の要求はとどまるところを知らず、高速化への要求と期待は高まるばかりである。このため、デバイス面からは種々の高速デバイスに関する研究が進められている。また、アーキテクチャ面では並列処理による高速化を目差し、様々なレベルにおいて並列性をいかすためのシステムの提案、研究が盛んに行われている。

本論文の目的は、SISD (Single Instruction Stream and Single Data Stream) なるノイマン型計算機の機構を離れて、演算レベルにおいて並列性を発揮し、制御の面からも高速演算を助長する演算実行方式を提案し、理論的かつ実際の立場で、新たな計算機アーキテクチャのための1つの方法論を展開することにある。

第2章 高速演算方式

本章では、まず、従来の計算機における代表的な高速演算方式について述べ、並列演算方式計算機の代表的な例について概観した。次に、本論文で考察するフィードフォワード型演算方式およびハードウェアマッピング型演算方式について概説した。

フィードフォワード型演算方式は、従来行われている並列演算方式とは異なり、演算の副作用を積極的に用いて高速演算を行う方式である。またハードウェアマッピング型演算方式は、処理プログラムをハードウェア化したようにマッピングすることで高速演算を行う方式である。

両者に共通していることは、これらの演算方式を用いた計算機を構築した場合、制御装置が処理を行う機械語命令には『演算命令が存在しない』ということである。制御装置はこれらの演算方式を用いた処理装置に対して演算データを転送するだけで、所望の演算結果を得ることができる。

第3章 フィードフォワード計算機

本章では、フィードフォワード型演算方式について、種々の特性を解析した。従来の計算機の演算装置は、1個の演算回路と数～数十個のレジスタで構成されている。このため演算命令は、(i)どのデータに対し、(ii)どのような演算を施し、(iii)どこに格納するか、の3種類を指定する必要がある（このうち、いくつかがデフォルト値となっているものもある）。これに対し、フィードフォワード型演算方式を用いたフィードフォワード計算機の演算装置は、複数の機能を固定した演算回路と、各演算回路に対して数～数千ステップ以前までの演算結果をすべて格納しておくレジスタ群から構成されている。一対の演算データが入力されると、すべての演算回路が動作し、それぞれの演算結果はレジスタ群に格納される。したがって、フィードフォワード計算機においては、演算の種類、演算結果の格納先の指定は不要となり、演算を行うデータを指定するだけでよく、演算命令は隠れたものになる。また、複数の演算回路が同時に動作しているため、同じ演算データに対し異なる演算（例えば、 $A+B$ と $A-B$ ）を行う場合、後者（ $A-B$ ）の演算結果はすでに前者（ $A+B$ ）の演算実行時に得られているため、後者（ $A-B$ ）を実行する命令は不要となる。これは、従来のノイマン型計算機で実行するプログラムに対し、プログラムステップ数を短縮できることを意味している。さらに、数～数千ステップ以前までの演算結果がすべてレジスタ群に格納されているため、ノイマン型計算機に比べ、バスライン上を流れるデータ数が減少し、フォン・ノイマン・ボトルネックがある程度緩和できる。

以上をまとめると、ノイマン型計算機に比べ本フィードフォワード計算機は、同じプログラムを実行する場合、

- i) 命令数が減少する。
- ii) バスライン上のデータ転送量が減少する。

等の利点を持ち、スループットの向上が期待できる。本章では、まず、実際に試作を行ったフィードフォワード計算機について、その具体的なハードウェア／ソフトウェア構成と、性能および問題点を検討した。その結果本試作機は、サンプルプログラムにおいては、マイクロプロセッサMC68000の約2倍の性能が得られた。さらに理論的観点から、最小命令実行時間の推定をペトリネッ

トを用いて行った。その結果、最適制御を行うと、現在の試作機よりも40～62%命令実行時間が短縮できることがわかった。

次に、フィードフォワード型演算方式の欠点である演算時間のばらつきを無くするため、各演算装置をパイプライン化した場合について、パイプラインの乱れによるアイドル時間に関する考察を行った。その結果、平均アイドル時間は、プログラムの性質によって大きく影響を受け、コンパイラの最適化フェーズが重要であることを示した。

また、フィードフォワード計算機特有の並列処理およびレジスタ構成によるプログラムステップ数の減少率について、理論的考察を行い、コンパイラに対する基本的方策を示した。これを基に、コンパイラにおける最適化手法を示した。さらに、フィードフォワード計算機向き計算アルゴリズムを提案し、幾つかの数値計算に関する計算量を評価した。その結果、多項式の評価に関しては、乗算回数を1/2程度にまで減少できることがわかった。最後に、命令のデコードビット数に関する検討を行い、デコード時間と演算時間の比によって最適なデコードビット数が存在することを示した。また、本フィードフォワード計算機から派生したマルチオペレーション計算機に関して、その構成を示し、コンパイラおよびシュミレータを用いて検討を行った。その結果、十分な個数の演算装置とメモリバンクを用意すると、従来のノイマン型計算機よりも、内積の計算で約2.8倍、FFTの計算で約1.7倍の実行速度が得られることがわかった。

第4章 プログラム可能論理演算装置

本章では、ハードウェアマッピング型演算方式について種々の特性を解析した。あるプログラムが与えられたとき、そのプログラムを最も高速に処理する方法の一つに、要求される処理をすべてハードウェア化する方法がある。この方法は、種々の専用機上に実現されており、画像処理やFFT演算等、特定の処理に関しては確かに高速である。しかし、他の処理については、従来のノイマン型計算機と同様に1命令ずつ実行するため、高速であることは言えない。ハードウェアマッピング型演算方式の一実現方法であるプログラム可能論理演算装置は、ある程度処理内容は限定するが、あたかも必要とされる処理がハードウェア化されたかのように、その処理をプログラム可能論理演算装置上にマッピングすることができる。

したがって、一度プログラム可能論理演算装置上に処理をマッピングしたならば、入力データをプログラム可能論理演算装置へ転送すれば、処理に応じた遅延時間の後に処理結果を得ることができる。このため、プログラム可能論理演算装置はデータの記憶機能、演算機能およびデータの転送機能を含む記憶・処理装置となる。このことは従来のノイマン型計算機において、処理を行うプログラムステップのフェッチ・デコードといった逐次の動作が、PLUへのプログラムロードという一括動作で置き換えられることを意味している。さらに処理の実行についても、ノイマン型計算機における逐次実行に対して、PLUでは入力データの決定した処理からその処理を開始するため高速演算が期待できる。

以上、PLUの特徴をまとめると、

- 1) プログラム実行時のデータ転送量が少なく、フォン・ノイマン・ボトルネックの緩和が可能

である。

- ii) プログラム実行時は入力データの確定した処理から実行が開始されるため、高速・並列演算が可能である。

本章では、まず、CPUに対する演算処理の負荷を減少させることのできるプログラム可能論理演算装置の概念および一般的な構成について述べた。次に、本論文で考察するプログラム可能論理演算装置をそのハードウェア構成によって4種類（内部構成に関して2種類，データ転送方式に関して2種類）に分類した。また，その中の一構成方式であるプログラム可能論理演算装置について，実際に試作を行い，その具体的構成，基本動作特性および問題点を示した。その結果，本試作機は1ステップあたり20～110 ns程度で演算を行えることを示した。次に，プログラム可能論理演算装置に対する機能割り当てに関する検討を行い，数式のマッピングおよびバスの割り当てアルゴリズムを与えた。さらに，マッピングの最適化方法について検討し，数式パターンの数え上げ問題を通してその有効性を示した。次に，数式の処理時間およびデータ転送量に関する検討を加え，ノイマン型計算機との比較を行った。さらに，プログラムの実行時間について考察を行い，プログラム可能論理演算装置を有効利用できるプログラム特性を明らかにし，プログラム可能論理演算装置を用いた計算機の位置付けを行った。最後に，本来メモリ装置であるプログラム可能論理演算装置に対し，従来のメモリ装置と同じく仮想化の技術を適用し，その有効性に関して理論的な考察を行った。

第5章 結 論

本章は，本論文を要約し，主要な結果を総括した。

審 査 結 果 の 要 旨

電子計算機の普及に伴い情報処理分野が拡大し、より高速な情報処理システムの開発が強く望まれている。しかしながら従来の逐次型システムでは処理が逐次的に行われるため十分な処理速度が望めない。これに対処するためには並列処理の機能を持った情報処理システムを開発する必要がある。本論文は、この立場の下で、演算レベルにおいて並列演算を制御し、処理するシステムを提案し、実際のおよび理論的立場から本システムの有効性を示したもので、全編5章よりなる。

第1章は緒論であり、本論文の背景および本研究の動機について述べている。

第2章では、従来の逐次型処理システムで行われている高速演算方式の背景と問題点について述べている。そして従来の高速演算方式と異なるフィードフォワード型演算方式およびハードウェアマッピング型演算方式の処理概念を示し、これら2つの方式と従来の方式との違いを明らかにしている。

第3章では、試作したフィードフォワード型演算方式を用いた計算機について、その動作概念とこの計算機の構成法を述べている。さらに、命令実行時間や性能などについて理論的考察を行うと共に、この計算機の処理に適した計算アルゴリズムを示し、コンパイラにおける最適化問題の解答を与えている。ついで、サンプルプログラムを用いて試作システムの性能を評価し、同一のクロックを持つマイクロプロセッサMC 68000の性能の約2倍であること、多項式数値計算においては乗算回数を1/2程度減少できることなどの具体的な結果を得ている。これらの結果は、フィードフォワード型演算方式の効果を示すものであり、非常に興味ある成果である。

第4章では、ハードウェアマッピング型演算方式を実現するプログラム可能論理演算装置(PLU)について、その動作概念およびシステムの構成法を示している。ついで、PLUシステムを試作し、演算遅延時間等の基本的な動作パラメータを得ている。また、逆ポーランド記法で記述された数式をハードウェア上に直接マッピングして処理できる有効な方式を示している。さらに、理論的見地から逐次型計算機と比較してこの数式処理の処理時間を1/9程度にまで短縮できることなどの結果を得ている。このPLUは、時間的・空間的並列性を十分引き出している汎用高速処理装置として重要な知見である。

第5章は結論である。

以上要するに本論文は、従来の逐次型計算機の考え方から離れ、演算の並列性を利用して計算機の高速化を実現する方法を提案し、試作を経て性能評価したものであり、今後の情報処理システム構築上新たな分野を開くもので情報工学および計算機工学の発展に寄与するところが少なくない。

よって、本論文は工学博士の学位論文として合格と認める。